Faster Convex Optimization: Simulated Annealing with an Efficient Universal Barrier

Jacob Abernethy, Elad Hazan

July 20, 2016

Outline

Outline

A Simulated Annealing

1. Physical interpretation
2. General framework for discrete problems
3. Extension to convex optimization
4. Hit-and-run-algorithm

B Interior-points method

1. Basics
2. Entropic barrier

Outline

A Simulated Annealing

1. Physical interpretation
2. General framework for discrete problems
3. Extension to convex optimization
4. Hit-and-run-algorithm

B Interior-points method

1. Basics
2. Entropic barrier

C Results

1. Improved simulated annealing
2. Consequences

Spoiler

Central path (Entropic barrier) $= \mathbb{E}$[Simulated Annealing]

Annealing: heating then (slowly) cooling a material to increase its ductility and reduce its hardness.

Steels with high ductility, low hardness: Typical when the molecular structure has **low potential energy**

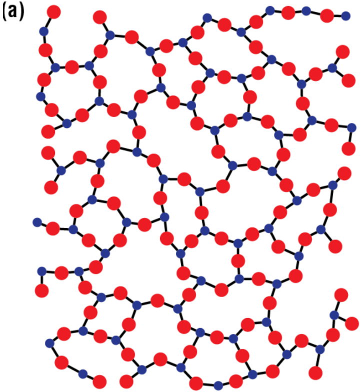


Figure: (a) Initial state (stable). (b) After annealing

Idea of annealing:

1. Take some material, like steel
2. Heat the material at high temperature
3. Cool down slowly the material
4. ????
5. Profit!

Idea of annealing:

1. Take some material, like steel
2. Heat the material at high temperature
3. Cool down slowly the material
4. ????
5. Profit!

Why does it works?

Boltzmann distribution

$$P(\text{state transition}) \propto e^{-\frac{E(\text{State})}{T}}$$

Where

- E is the energy of the next state
- T is the temperature

Boltzmann distribution

$$P(\text{state transition}) \propto e^{-\frac{E(\textbf{State})}{T}}$$

Recall: **minimizing** energy

- High T: Can jump to high-energy state more easily
- Low T: Tend to be greedy (more weight for low E)

Boltzmann distribution

$$P(\text{state transition}) \propto e^{-\frac{E(\textbf{State})}{T}}$$

Recall: **minimizing** energy

- High T: Can jump to high-energy state more easily
- Low T: Tend to be greedy (more weight for low E)

Simulated annealing for discrete problems: Set $state = x$ and $E(state) = f(x)$.
At iteration k,

1. Choose a temperature t_k
2. Define a (small) set of neighbors $\mathcal{S}$
3. Sample a point x in S where $P(x = x_i) = \dfrac{\exp(-f(x_i)/t_k)}{\sum_j \exp(-f(x_j)/t_k)}$

Simulated annealing for continuous convex problem

General formulation, for $\mathcal{X}$ convex.

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ s.t. \quad & x \in \mathcal{X} \end{aligned}$$

Assume $\|\mathcal{X}\|_2 < R$. Let $c_k = \frac{c}{t_k}$.

Simulated annealing for continuous convex problem

General formulation, for $\mathcal{X}$ convex.

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & c^T x \\ s.t. \quad & x \in \mathcal{X} \end{aligned}$$

Assume $\|\mathcal{X}\|_2 < R$. Let $c_k = \frac{c}{t_k}$.

Boltzmann's distribution: $P_{c_k}(x) = \dfrac{\exp(-c_k^T x)}{\int_{\mathcal{X}} \exp(-c_k^T x) dx}$, but $\int_{\mathcal{X}} = \mathcal{O}(2^n)$ in general.

Approximation at point x_k: (Algorithm `HitAndRun`)

1. Take random direction $u \sim \mathcal{N}(0, \Sigma_k)$, Σ_k is an estimate of the covariance matrix at x_k
2. Determine line segment $\ell_k = \{x_k + \alpha u_k,\ \alpha \in \mathbb{R}\} \cap \mathcal{X}$ (using line-search).
3. Sample a point x_{k+1} following $P_{c_k}(x)$ restricted to ℓ_k.

Algorithm `SimulatedAnnealing` using warm restart of `HitAndRun`:
Use $n+1$ different paths,

- One for the solution (x_k)
- The n others (y_k^j) are for estimating covariance Σ_k

Algorithm `SimulatedAnnealing` using warm restart of `HitAndRun`:
Use $n+1$ different paths,

- One for the solution (x_k)
- The n others (y_k^j) are for estimating covariance Σ_k

Algorithm `SimulatedAnnealing` (*Kalai, Vempala [2006]*):

- Temperature's law: $t_k = (1 - \frac{1}{\sqrt{n}})^k R$, $c_k = \frac{c}{t_k}$
- $x_{k+1} = \texttt{HitAndRun}(x_k, \Sigma_k, \mathcal{X}, c_k, N)$ (N = `HitAndRun` iterations)
- Σ_{k+1} is estimated with n vectors $y_{k+1}^j = \texttt{HitAndRun}(y_k^j, \Sigma_k, \mathcal{X}, c_k, N)$
- Until $k = \mathcal{O}(\sqrt{n}\log(n/\epsilon))$ (required for ϵ-solution)

Algorithm SimulatedAnnealing using warm restart of HitAndRun:
Use $n+1$ different paths,

- One for the solution (x_k)
- The n others (y_k^j) are for estimating covariance Σ_k

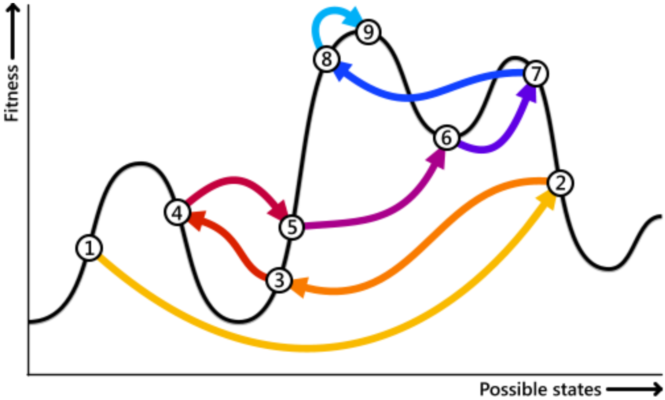
Algorithm SimulatedAnnealing (*Kalai, Vempala [2006]*):

- Temperature's law: $t_k = (1 - \frac{1}{\sqrt{n}})^k R$, $c_k = \frac{c}{t_k}$
- $x_{k+1} = \texttt{HitAndRun}(x_k, \Sigma_k, \mathcal{X}, c_k, N)$ (N = HitAndRun iterations)
- Σ_{k+1} is estimated with n vectors $y_{k+1}^j = \texttt{HitAndRun}(y_k^j, \Sigma_k, \mathcal{X}, c_k, N)$
- Until $k = \mathcal{O}(\sqrt{n}\log(n/\epsilon))$ (required for ϵ-solution)

Works only if $P_{c_k} \approx P_{c_{k+1}}$, satisfied if

- t decreases in $(1 - \frac{1}{\sqrt{n}})^k$
- $N = \mathcal{O}(n^3)$

Complexity: $\sqrt{n}\log(n/\epsilon) \times n^3 \times n = \mathcal{O}(n^{4.5}\log(n))$


Fitness
Possible states
1
2
3
4
5
6
7
8
9

Interior-points method and barrier function

Idea: replace

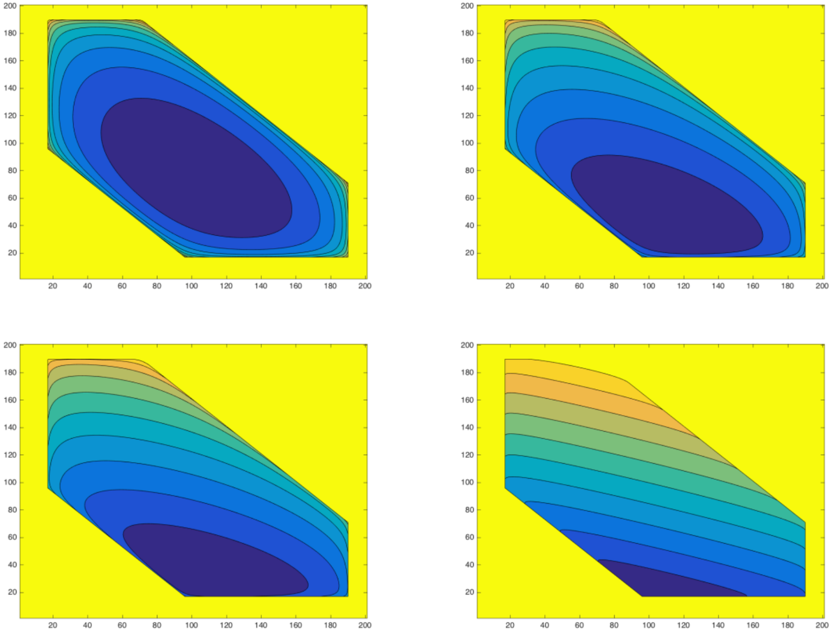
$$\begin{aligned} \min_x \quad & c^T x \\ s.t. \quad & x \in \mathcal{X} \end{aligned}$$

by successive approximations x_k solving (with Newton's method)

$$\min_x \beta_k c^T x + F(x) \quad , \qquad \beta_k = \left(1 + \frac{1}{\sqrt{\mu}}\right)^k$$

Complexity: $\mathcal{O}\left(\sqrt{\nu}\log(\nu/\epsilon)\right) \times \mathcal{O}(n^3)$.

Remark: Works only if β_k grows slowly


200 180 160 140 120 100 80 60 40 20
20 40 60 80 100 120 140 160 180 200
200 180 160 140 120 100 80 60 40 20
20 40 60 80 100 120 140 160 180 200
200 180 160 140 120 100 80 60 40 20
20 40 60 80 100 120 140 160 180 200
200 180 160 140 120 100 80 60 40 20
20 40 60 80 100 120 140 160 180 200

Universal barrier for convex sets

Interior point methods work using *self-concordant barrier* for set $\mathcal{X}$.

Self-concordant function: A nice function for Newton's method.

Theorem (*Bubeck, Eldan [2014]*): The function $u^*_{\mathcal{K}}$ is a self-concordant barrier for the convex set $\mathcal{K}$, with parameter $\nu = n(1 + o(1))$:

$$u^*_{\mathcal{K}}(x) = \sup_{\theta \in \mathbb{R}^n} \theta^T x - u_{\mathcal{K}}(\theta) \qquad ; \qquad u_{\mathcal{K}}(\theta) = \log \int_{y \in \mathcal{K}} \exp(\theta^T y) dy$$

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u_{\mathcal{K}}^*(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u^*_{\mathcal{K}}(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y)dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u^*_{\mathcal{K}}(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y)dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Proof:

1. $\mathbb{E}_{x\sim P_c}[x] = -\nabla A(c)$ Property from exponential family

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u^*_{\mathcal{K}}(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y)dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Proof:

1. $\mathbb{E}_{x\sim P_c}[x] = -\nabla A(c)$ Property from exponential family
2. $-\nabla A(c) = -\arg\max_{x\in \mathrm{dom}(A^*)} c^T x - A^*(x)$ Fenchel conjugate

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta > 0} \arg\min \beta c^T x + u^*_{\mathcal{K}}(x)$
- Heat path: $\bigcup_{t > 0} \mathbb{E}_{x \sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y) dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Proof:

1. $\mathbb{E}_{x \sim P_c}[x] = -\nabla A(c)$ Property from exponential family
2. $-\nabla A(c) = -\arg\max_{x \in \mathbf{dom}(A^*)} \; c^T x - A^*(x)$ Fenchel conjugate
3. $-\nabla A(c) = \arg\min_{x \in \mathbf{dom}(A^*)} \; -c^T x + A^*(x)$

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u_{\mathcal{K}}^*(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y) dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Proof:

1. $\mathbb{E}_{x\sim P_c}[x] = -\nabla A(c)$ Property from exponential family
2. $-\nabla A(c) = -\arg\max_{x\in \text{dom}(A^*)}\ c^T x - A^*(x)$ Fenchel conjugate
3. $-\nabla A(c) = \arg\min_{x\in \text{dom}(A^*)}\ -c^T x + A^*(x)$
4. We can show that $\text{dom}(A^*) = -\mathcal{X}$

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u^*_{\mathcal{K}}(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y)dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Proof:

1. $\mathbb{E}_{x\sim P_c}[x] = -\nabla A(c)$ Property from exponential family
2. $-\nabla A(c) = -\arg\max_{x\in \mathrm{dom}(A^*)}\ c^T x - A^*(x)$ Fenchel conjugate
3. $-\nabla A(c) = \arg\min_{x\in \mathrm{dom}(A^*)}\ -c^T x + A^*(x)$
4. We can show that $\mathrm{dom}(A^*) = -\mathcal{X}$
5. $\nabla A(c) = \arg\min_{x\in -\mathcal{X}}\ -c^T x + A^*(x)$

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u^*_{\mathcal{K}}(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y)dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Proof:

1. $\mathbb{E}_{x\sim P_c}[x] = -\nabla A(c)$ Property from exponential family
2. $-\nabla A(c) = -\arg\max_{x\in \mathsf{dom}(A^*)}\ c^T x - A^*(x)$ Fenchel conjugate
3. $-\nabla A(c) = \arg\min_{x\in \mathsf{dom}(A^*)}\ -c^T x + A^*(x)$
4. We can show that $\mathsf{dom}(A^*) = -\mathcal{X}$
5. $\nabla A(c) = \arg\min_{x\in -\mathcal{X}}\ -c^T x + A^*(x)$
6. $\nabla A(c) = \arg\min_{x\in \mathcal{X}}\ c^T x + A^*(-x)$

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u^*_{\mathcal{K}}(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y) dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Proof:

1. $\mathbb{E}_{x\sim P_c}[x] = -\nabla A(c)$ Property from exponential family
2. $-\nabla A(c) = -\arg\max_{x\in \text{dom}(A^*)} \; c^T x - A^*(x)$ Fenchel conjugate
3. $-\nabla A(c) = \arg\min_{x\in \text{dom}(A^*)} \; -c^T x + A^*(x)$
4. We can show that $\text{dom}(A^*) = -\mathcal{X}$
5. $\nabla A(c) = \arg\min_{x\in -\mathcal{X}} \; -c^T x + A^*(x)$
6. $\nabla A(c) = \arg\min_{x\in \mathcal{X}} \; c^T x + A^*(-x)$
7. However $A^*(-x) = u^*_{\mathcal{X}}(x)$ $\Leftrightarrow A(x) = u_{\mathcal{X}}(-x)$

Link between "Heat Path" and "Central Path"

- Central path: $\bigcup_{\beta>0} \arg\min \beta c^T x + u^*_{\mathcal{K}}(x)$
- Heat path: $\bigcup_{t>0} \mathbb{E}_{x\sim P_{c/t}(x)}[x]$

Idea: (assume $t = \beta = 1$)

- $A(c) = \log \int_{\mathcal{X}} \exp(-c^T y) dy$ Equal to $u_{\mathcal{X}}(-c)$
- $P_c(x) = \exp(-c^T x - A(c))$ Exponential family

Proof:

1. $\mathbb{E}_{x\sim P_c}[x] = -\nabla A(c)$ Property from exponential family
2. $-\nabla A(c) = -\arg\max_{x\in \mathbf{dom}(A^*)} \; c^T x - A^*(x)$ Fenchel conjugate
3. $-\nabla A(c) = \arg\min_{x\in \mathbf{dom}(A^*)} \; -c^T x + A^*(x)$
4. We can show that $\mathrm{dom}(A^*) = -\mathcal{X}$
5. $\nabla A(c) = \arg\min_{x\in -\mathcal{X}} \; -c^T x + A^*(x)$
6. $\nabla A(c) = \arg\min_{x\in \mathcal{X}} \; c^T x + A^*(-x)$
7. However $A^*(-x) = u^*_{\mathcal{X}}(x)$ $\Leftrightarrow A(x) = u_{\mathcal{X}}(-x)$
8. $\arg\min_{x\in \mathcal{X}} \; c^T x + u^*_{\mathcal{X}}(x)$ = Central Path

Consequences for `SimulatedAnnealing` algorithm:

- New temperature schedule: $t_k = (1 - \frac{1}{\sqrt{n}})^k \rightarrow t_k = (1 - \frac{1}{\sqrt{\nu}})^k$
- New complexity: $\mathcal{O}(n^{4.5}) \rightarrow \mathcal{O}(\sqrt{\nu} n^4)$
- $\nu = \mathcal{O}(n)$
- Randomized version of interior-point algorithms
  - Does not require the computation of the barrier
  - No gradient/Hessian needed
  - Higher complexity (factor of $\mathcal{O}(n)$)
  - Line-search for estimating ℓ_k
- Main assumption: oracle $x \in \mathcal{X}$ is cheap